On Learning Linear Functions from Subset and Its Applications in Quantum Computing

Gábor Ivanyos
Institute for Computer Science and Control, Hungarian Academy of Sciences, Budapest, Hungary
Gabor.Ivanyos@sztaki.mta.hu

Anupam Prakash
CNRS, IRIF, Université Paris Diderot 75205 Paris, France
anupam@irif.fr

Miklos Santha
CNRS, IRIF, Université Paris Diderot 75205 Paris, France; and
Centre for Quantum Technologies, National University of Singapore, Singapore 117543
miklos.santha@gmail.com

Abstract

Let $\mathbb{F}_q$ be the finite field of size q and let $\ell : \mathbb{F}_q^n \to \mathbb{F}_q$ be a linear function. We introduce the *Learning From Subset* problem $\mathrm{LFS}(q,n,d)$ of learning ℓ, given samples $u \in \mathbb{F}_q^n$ from a special distribution depending on ℓ: the probability of sampling u is a function of $\ell(u)$ and is non zero for at most d values of $\ell(u)$. We provide a randomized algorithm for $\mathrm{LFS}(q,n,d)$ with sample complexity $(n+d)^{O(d)}$ and running time polynomial in $\log q$ and $(n+d)^{O(d)}$. Our algorithm generalizes and improves upon previous results [8, 10] that had provided algorithms for $\mathrm{LFS}(q,n,q-1)$ with running time $(n+q)^{O(q)}$. We further present applications of our result to the *Hidden Multiple Shift* problem $\mathrm{HMS}(q,n,r)$ in quantum computation where the goal is to determine the hidden shift s given oracle access to r shifted copies of an injective function $f : \mathbb{Z}_q^n \to \{0,1\}^l$, that is we can make queries of the form $f_s(x,h) = f(x-hs)$ where h can assume r possible values. We reduce $\mathrm{HMS}(q,n,r)$ to $\mathrm{LFS}(q,n,q-r+1)$ to obtain a polynomial time algorithm for $\mathrm{HMS}(q,n,r)$ when $q = n^{O(1)}$ is prime and $q-r = O(1)$. The best known algorithms [5, 8] for $\mathrm{HMS}(q,n,r)$ with these parameters require exponential time.

2012 ACM Subject Classification Theory of computation → Quantum computation theory

Keywords and phrases Learning from subset, hidden shift problem, quantum algorithms, linearization

Digital Object Identifier 10.4230/LIPIcs.ESA.2018.66

Related Version A full version of the paper is available at [11], `https://arxiv.org/abs/1710.02581`.

Funding A part of the research was accomplished while the first two authors were visiting the Centre for Quantum Technologies (CQT), National University of Singapore. The research at CQT was partially funded by the Singapore Ministry of Education and the National Research Foundation under grant R-710-000-012-135. This research was supported in part by the QuantERA ERA-NET Cofund project QuantAlgo and by the Hungarian National Research, Development and Innovation Office – NKFIH, Grant K115288.

26th Annual European Symposium on Algorithms (ESA 2018).
Editors: Yossi Azar, Hannah Bast, and Grzegorz Herman; Article No. 66; pp. 66:1–66:14

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

1.1 Learning with noise

Let $n \geq 1$ and $q > 1$ be integers. We denote by $\mathbb{Z}_q$ the ring of integers modulo q, and by $\mathbb{F}_q$ the finite field on q elements, when q is some power of a prime number. When q is prime then $\mathbb{Z}_q$ coincides with $\mathbb{F}_q$, and we will use the notation $\mathbb{F}_q$. Let $\ell : \mathbb{F}_q^n \to \mathbb{F}_q$ be an n-variable linear function. The main subject of this paper is to learn ℓ given partial information about the values $\ell(u)$ for uniformly random samples u from $\mathbb{F}_q^n$. In the ideal setting, when we have access to the values $\ell(u)$ for uniformly random samples from $\mathbb{F}_q^n$, the problem is canonical and perfectly understood: after getting n independent samples, we can determine ℓ by Gaussian elimination in polynomial time. But when instead of the exact values we receive only some property satisfied by them, the problem can become much more difficult.

Since an element of $\mathbb{F}_q^n$ can be specified with $n \log q$ bits, we will say that *an algorithm is in polynomial time* if it runs in time polynomial in both n and $\log q$. Let $f(n, q)$ be a function of n and q, then we say that a function $g(n, q) \in \widetilde{O}(f)$ if $g(n, q) \leq f(n, q) \log^c(nq)$ for some constant c for sufficiently large n and q. By the *sample complexity* of an algorithm we mean the number of samples used by it.

There is a somewhat similar context to the learning model we investigate, it is the model where the values $\ell(u)$ are perturbed by some random noise. The first example of such a work is by Blum et al. [3] on the *Learning Parity with Noise* problem $\mathrm{LPN}(n, \eta)$, where $\eta < 1/2$. Here we have access to tuples $(u, b) \in \mathbb{F}_2^n \times \mathbb{F}_2$, where u is a uniformly random element of $\mathbb{F}_2^n$ and $b = \ell(u) + e$, where e is a random 0–1 variable with $\Pr[e = 1] = \eta$. For constant noise rate $0 < \eta < 1/2$, the best known algorithm for $\mathrm{LPN}(n, \eta)$ is from [3]. It has both sample and time complexity of $2^{O(n/\log n)}$, and therefore only marginally beats the trivial exhaustive search algorithm of complexity $2^{O(n)}$.

The *Learning With Error* problem $\mathrm{LWE}(q, n, \chi)$ is a generalization by Regev [17] of LPN to larger fields. Here q can be any prime number, and χ is a probability distribution on $\mathbb{F}_q$. Similar to LPN, we have access to tuples $(u, b) \in \mathbb{F}_q^n \times \mathbb{F}_q$, where u is a uniformly random element of $\mathbb{F}_q^n$ and $b = \ell(u) + e$, with the random variable e having distribution χ. Under the assumptions that q is bounded by some polynomial function of n, and that $\chi(0) \geq 1/q + 1/p(n)$, for some polynomial p, the problem can be solved classically with sample and time complexity $2^{O(n)}$. The case when $\chi = \Psi_\alpha$, the discrete Gaussian distribution of standard deviation αq, is of particular interest for lattice based cryptography. Indeed, one of the main results of [17] is that for appropriate parameters, solving $\mathrm{LWE}(q, n, \Psi_\alpha)$ is at least as hard as quantumly solving several cryptographically important lattice problems in the worst case. In a subsequent work a classical reduction of some of these lattice problems to LWE was given by Peikert [15].

In [2] Arora and Ge introduced a more structured noise model for learning linear functions over $\mathbb{F}_2^n$. In the *Learning Parity with Structured Noise* problem $\mathrm{LPSN}(n, m)$ the samples arrive in groups of size m, that is in one sampling step we receive $(u_1, b_1), \ldots, (u_m, b_m)$, where $(u_i, b_i) \in \mathbb{F}_2^n \times \mathbb{F}_2$, for $i = 1, \ldots, m$. Here $u_1, \ldots, u_m$ are independent random elements drawn from $\mathbb{F}_2^n$, and $b_i = \ell(u_i) + e_i$, where the the noise vector $e = (e_1, \ldots, e_m) \in \mathbb{F}_2^m$ must have Hamming weight less than $m/2$. The chosen noise vector e can depend on the sample $(u_1, \ldots, u_m)$, but the model has an important restriction (structure) compared to the previous error models. Since the Hamming weight of e is less than $m/2$, it is guaranteed that in every sampling group the majority of the bits b_i is correct, that is coincides with $\ell(u_i)$. In fact, the model of Arora and Ge is somewhat more general. Let P be any m-variable polynomial over $\mathbb{F}_2^m$, for which there exists $a \in \mathbb{F}_2^m$, such that $a \neq c + c'$ for all $c, c' \in \mathbb{F}_2^m$

satisfying $P(c) = P(c') = 0$. Then the error vector can be any $e \in \mathbb{F}_2^m$ satisfying $P(e) = 0$. The main result of [2] is that LPSN(n, m) can be solved in time $n^{O(m)}$, implying that the linear function can be learnt in polynomial time when m is constant.

1.2 Learning from subset

We consider here a different model of learning linear functions where the difficulty doesn't come from the noisy sampling process, but from the fact that instead of obtaining the actual values of the sampled elements, we only receive some partial information about them.

Such a model was first considered by Friedl et al. [8] with the *Learning From Disequations* problem LFD(q, n) where q is a prime number. Here we never get sample elements from the kernel of ℓ, that is we can only sample u if $\ell(u) \neq 0$, which explains the name of the problem. Friedl et al. [8] consider distributions p which are not necessarily uniform on their support, in fact they only require that $p(u) = p(v)$ whenever $\ell(u) = \ell(v)$.

The reason to consider this learning problem in [8] is that the *Hidden Shift* problem HS(q, n), a paradigmatic problem in quantum computing, can be reduced in quantum polynomial time to LFD(q, n). In HS(q, n) we have oracle access to two injective functions f_0 and f_1 over $\mathbb{F}_q^n$ with the promise that for some element $s \in \mathbb{F}_q^n$, we have $f_1(x) = f_0(x - s)$, for all $x \in \mathbb{F}_q^n$. The element s is called the *hidden shift*, and the task is to find it. It is proven in [8] that LFD(q, n) can be solved in time $(n+q)^{O(q)}$. This result implies that there exists a quantum algorithm for HS(q, n) of similar complexity. When q is constant, these algorithms are therefore polynomial time.

In a subsequent paper [10] Ivanyos extended the work of [8] to the case when q is a prime power, both for LFD(q, n) and HS(q, n). The complexity bounds obtained are very similar to the bounds of [8], and therefore his results imply that LFD(q, n) can be solved in polynomial time, and that HS(q, n) in quantum polynomial time when q is a prime power of constant size.

Observe that the complexity bound $(n + q)^{O(q)}$ is not only not polynomial in $\log q$, but is not even exponential, in fact it is doubly exponential. Therefore [8] and [10] not only leave open the question whether, in general, it is possible to obtain a (quantum) algorithm for LFD(q, n) and HS(q, n) with running time polynomial in n and q, but also the question of the existence of algorithms which have running time polynomial in n and $\log q$. These questions are still open today.

In this work we introduce a generalization of the learning problem LFD. While in LFD the sampling distribution had to avoid the kernel of ℓ, in our model the input contains a set $A \subseteq \mathbb{F}_q$, and we sample from distributions whose support contains only those elements u, for which $\ell(u) \in A$. As in [8], we require that the elements with the same ℓ-value have identical probabilities. We allow these probabilities to be exponentially small and even 0.

▶ **Definition 1.** Let $A \subset \mathbb{F}_q$, where q is a prime power, let $\ell : \mathbb{F}_q^n \to \mathbb{F}_q$ be a linear function, and let p be a distribution over $\mathbb{F}_q^n$. We say that the ℓ-*image* of p is A if $\ell(\text{supp}(p)) = A$. The distribution is ℓ-*symmetric* if $\ell(u) = \ell(v)$ implies $p(u) = p(v)$. If the ℓ-image of p is a subset of A and p is also ℓ-symmetric, we say that p is an (A, ℓ)-*distribution*.

In other words, p is an (A, ℓ)-distribution if p is constant on each affine subspace $V_\alpha = \{u \in \mathbb{F}_q^n : \ell(u) = \alpha\}$, for $\alpha \in \mathbb{F}_q$, and moreover p is zero on V_α, whenever $\alpha \notin A$. It is not hard to see that for $|A| < q$, if p is simultaneously an (A, ℓ)-distribution and an (A, ℓ')-distribution then ℓ' is a constant multiple of ℓ. On the other hand, non-zero constant multiples of a linear function can not be distinguished in general in this model: for example, if $A = \mathbb{F}_q \setminus \{0\}$, then for every $c \neq 0$, an (A, ℓ)-distribution is also an $(A, c\ell)$-distribution.

▶ **Definition 2.** The *Learning From Subset* problem $\mathrm{LFS}(q, n, d)$ is parametrized by three positive integers q, n and d, where q is a prime power and $2 \leq d \leq q-1$.
Input: A set $A \subset \mathbb{F}_q$ of cardinality d and a sequence of N samples $u_1, \ldots, u_N$ from an (A, ℓ)-distribution for some nonzero linear function $\ell : \mathbb{F}_q^n \to \mathbb{F}_q$.
Output: A non zero constant multiple of ℓ.

For $d < d'$, an $\mathrm{LFS}(q, n, d)$ instance is also an $\mathrm{LFS}(q, n, d')$ instance, therefore the problem is harder for bigger d. For $d = 1$ the problem is simple because it becomes a system of linear equalities which can be solved by Gaussian elimination. When $d = q$ we don't receive any information from the samples and it is impossible to identify the linear function. When $d = q - 1$ and $A = \mathbb{F}_q \setminus \{0\}$, the problem LFS specializes to LFD, in fact the latter is the hardest instance of the former.

The first main result of our paper is a randomized algorithm for $\mathrm{LFS}(q, n, d)$ whose complexity depends exponentially on d, but only polynomially on $\log q$. This result shows that the increase of information by reducing the size of the set A can indeed be algorithmically exploited. More precisely, we show that for a sample size N which is a sufficiently large polynomial of n^d, there exists a randomized algorithm which in time polynomial in n^d and $\log q$, with probability $1/2$, determines ℓ up to a constant factor.

▶ **Theorem 3.** *There is a randomized algorithm for* $\mathrm{LFS}(q, n, d)$ *with sample complexity* $(n+d)^{O(d)}$ *and running time polynomial in* $\log q$ *and* $(n+d)^{O(d)}$.

The main interest of this result is that for constant d it gives a polynomial time algorithm for LFS. For $d = q - 1$ and $A = \mathbb{F}_q \setminus \{0\}$ it yields the same complexity bound as [8] and [10]. But observe, that even for non constant $d = o(q)$, it is asymptotically faster than the algorithms in the above papers.

1.3 Hidden multiple shifts

The original motivation for [8] to study LFD was its connection to the hidden shift problem. This problem was implicitly introduced by Ettinger and Høyer [7], while studying the dihedral hidden subgroup problem. The hidden shift problem can be defined in any group G. We are given two injective functions f_0 and f_1 mapping G to some arbitrary finite set. We are promised that for some element $s \in G$, we have $f_1(xs) = f_0(x)$, for every $x \in G$, and the task is to find s. As shown in [7], when G is abelian, the hidden shift in G is quantum polynomial time equivalent to the hidden subgroup problem in the semidirect product $G \rtimes \mathbb{Z}_2$. In the semidirect product the group operation is defined as $(x_1, b_1).(x_2, b_2) = (x_1 + (-1)^{b_1} x_2, b_1 + b_2)$, and the function $f(x, b) = f_b(x)$ hides the the subgroup $\{(0, 0), (s, 1)\}$. The quantum complexity of HS in the cyclic group $\mathbb{Z}_q$ (or equivalently, the complexity of the hidden subgroup in the dihedral group $\mathbb{Z}_q \rtimes \mathbb{Z}_2$) is a famous open problem in quantum computing. In [7] there is a quantum algorithm for this problem of polynomial quantum sampling complexity, but followed by an exponential time classical post-processing. The currently best known quantum algorithm is due to Kuperberg [13], and it is of subexponential complexity $2^{O(\sqrt{\log q})}$. Note that one could also consider shifts of non-injective functions. The extension of HS to such cases can become quite difficult even over $\mathbb{Z}_2^n$ where HS for injective functions is identical to the hidden subgroup problem. Results in this direction can be found e.g. in [9], [4] and [18].

As one could expect, the polynomial time algorithm for LFS with constant d has further consequences for quantum computing. Indeed, using this learning algorithm, we can solve in quantum polynomial time some instances of the hidden multiple shifts problem, a generalization of the hidden shift problem, which we define now.

For an element $s \in \mathbb{Z}_q^n$, a subset $H \subseteq \mathbb{Z}_q$ of cardinality at least 2, and a function $f : \mathbb{Z}_q^n \to \{0,1\}^l$, where l is an arbitrary positive integer, we define the function $f_s : \mathbb{Z}_q^n \times H \mapsto \{0,1\}^l$ as $f_s(x,h) = f(x - hs)$. We think about $f_s(x,h)$ as the hth *shift* of f by s. The task in the hidden multiple shift problem is to recover s when we are given oracle access, for some f and H, to f_s. This problem doesn't necessarily have a unique solution. Indeed, let us define $\delta(H,q)$ as the largest divisor of q such that $h - h'$ is divisible by $\delta(H,q)$ for every $h, h' \in H$. Pick $h_0 \in H$. Then for any $s' \in \frac{q}{\delta(H,q)}\mathbb{Z}_q^n$ and $h \in H$, we have $hs' = h_0 s' + (h - h_0)s' = h_0 s'$ whence $h(s + s') = hs + h_0 s'$ and therefore

$$f_{s+s'}(v,h) = f(v - h(s + s')) = f(v - h_0 s' - hs) = f'_s(v,h),$$

where $f'(v) = f(v - h_0 s')$. This means that s and $s + s'$ are indistinguishable by the set of shifts of f, and therefore we can only hope to determine (the coordinates of) s modulo $\frac{q}{\delta(H,q)}$. When q is a prime number, this problem of course doesn't arise.

▶ **Definition 4.** The *Hidden Multiple Shift* problem $\mathrm{HMS}(q,n,r)$ parametrized by three positive integers q, n and r, where $q > 1$ and $2 \le r \le q - 1$.
Input: A set $H \subseteq \mathbb{Z}_q$ of cardinality r.
Oracle input: A function $f_s : \mathbb{Z}_q^n \times H \to \{0,1\}^l$, where $s \in \mathbb{Z}_q^n$ and $f : \mathbb{Z}_q^n \to \{0,1\}^l$ is an injective function.
Output: $s \mod \frac{q}{\delta(H,q)}$.

The HMS problem was first considered by Childs and van Dam [5]. They investigated the cyclic case $n = 1$ and assumed that H is a contiguous interval and presented a polynomial time quantum algorithm for such an H of size $q^{\Omega(1)}$. Their result could probably be extended to constant n. However, for 'medium-size' n and q, such a result seems to be very difficult to achieve. Obtaining an efficient algorithm for medium sized n, q is also stated as an open problem [6], and it is noted that such a result would greatly simplify their algorithm. Intuitively, for small H the HMS appears to be 'too close' to the HS for which the so far best result is still what is given in [8].

For $r = q$, the HMS problem can be solved in quantum polynomial time. Indeed, in that case $H = \mathbb{Z}_q$, and $\mathbb{Z}_q^n \times H = \mathbb{Z}_q^{n+1}$ is an abelian group. The function f_s hides the subgroup generated by $(s,1)$, therefore we have an instance of the abelian hidden subgroup problem. When $r = 1$ the problem is void, there is no hidden shift. When $r = 2$, we have the standard hidden shift problem for which [8] and [10] gave a quantum algorithm of complexity $(n+q)^{O(q)} = (n+q)^{O(q+1-r)}$. Their method at a high level is a quantum reduction to (several instances of) $\mathrm{LFS}(q,n,q-1)$. These extreme cases suggest a strong connection between the classical complexity of $\mathrm{LFS}(q,n,d)$ and the quantum complexity of $\mathrm{HMS}(q,n,r)$ when $r = q + 1 - d$. Indeed, this turns out to be true. In our second main result we give a polynomial time quantum Turing reduction of $\mathrm{HMS}(q,n,r)$ to $\mathrm{LFS}(q,n,q+1-r)$, to obtain an algorithm of complexity $(n+q)^{O((q-r)^2)}$ for the former problem.

▶ **Theorem 5.** *Let q be a prime. Then there is a quantum algorithm which solves* $\mathrm{HMS}(q,n,r)$ *with sample complexity and in time* $(n+q)^{O((q-r)^2)}$.

The above Theorem yields a polynomial time algorithm for $\mathrm{HMS}(q,n,r)$ for the case when $q - r$ is constant and $q = n^{O(1)}$. We also present a Fourier sampling based algorithm for HMS which is polynomial time for a different set of parameters satisfying $\frac{r}{q} = 1 - \Omega(\frac{\log n}{n})$. We have the following result.

▶ **Theorem 6.** *There is a quantum algorithm that solves* $\mathrm{HMS}(q,n,r)$ *with high probability in time* $O(\mathrm{poly}(n)(\frac{q}{r})^{n+O(1)})$.

1.4 Our proof methods

The basic idea of the proof of Theorem 3 is a variant of linearization used in [8] and in [2], presented in the flavor of [10]. To give a high level description, observe that every u such that $p(u) \neq 0$ is a zero of the polynomial $f_{(A,\ell)}(x) = \prod_{a \in A}(\ell(x) - a)$. By Hilbert's Nullstellensatz, over the algebraic closure of $\mathbb{F}_q$, the polynomials which vanish on all the zeros of $f_{(A,\ell)}$ are multiples of $f_{(A,\ell)}$. In particular, every such polynomial which is also of degree at most d must be a scalar multiple of $f_{(A,\ell)}$. Interestingly, one could show that this consequence remains true with high probability, if we replace "all the zeros" by sufficiently many random samples provided that our (A, ℓ)-distribution is uniform (or nearly uniform) in the sense that $p(u)$ (the probability of sampling u) is the same (or almost the same) for every u such that $\ell(u) \in A$, independently on the actual value of $\ell(u)$. Therefore, in the (nearly) uniform case one could compute a nontrivial scalar multiple of $f_{(A,\ell)}$ by finding a nontrivial solution of a system of N homogeneous linear equations in $(n+d)^d$ unknowns (these are the coefficients of the various monomials in $f_{(A,\ell)}$). Then ℓ could be determined by factoring this polynomial. This method would be a direct generalization of the algorithms given in [8] and [10]. Indeed, in those papers one could just take $A = \mathbb{F}_q \setminus \{0\}$. However, the proofs (and in case of [10] even the algorithmic ingredients) are designed specially for small q and straightforward extensions would result in algorithms of complexity depending exponentially not only on d but on $\log q$ as well. Here we give an algorithm that depends polynomially on $\log q$ and that works without any assumption on uniformity. (In the case $A = \mathbb{F}_q \setminus \{0\}$ uniformity can actually be simulated by multiplying the sample vectors by random nonzero scalars.) Then, instead of divisibility by $f_{(A,\ell)}$ we prove that, with high probability, the polynomials that are zero on sufficiently many samples are divisible by $\ell(x) - a$ for the "most frequent" value $a \in A$. Then we find a scalar multiple of ℓ by factoring a nonzero polynomial from the space of those which are zeros on all the samples.

The subexponential LWE-algorithm of Arora and Ge [2] is based on implicitly solving a problem that can be cast as an instance of LFS where one of the coefficients of the linear function ℓ is known, $0 \in A$, and the (A, ℓ) distribution is such that 0 is the most likely value. More details are given in the full version [11].

The algorithm for solving $\mathrm{HMS}(q, n, r)$ in Theorem 5 is based on the following. After applying some standard preprocessing, we obtain samples of states that are projections to an r-dimensional space of $\mathrm{QFT}(|(u, s)\rangle)$ where QFT denotes the quantum Fourier transform on $\mathbb{Z}_q^n$, the vector $u \in \mathbb{Z}_q^n$ is sampled from the uniform distribution on $\mathbb{Z}_q^n$ and $(\cdot, \cdot)$ denotes the standard scalar product of $\mathbb{Z}_q^n$. If we are able to determine the scalar product (u, s) for n linearly independent u using the projected states, then s can also be computed using Gaussian elimination. However when q is not large enough compared to n then the error probability for computing (u, s) is too large and we get a system of noisy linear equations for which no efficient algorithms are known. Instead, we can devise a measurement, that at the cost of sacrificing a $1 - 1/q^{O(1)}$ fraction of the samples, yields samples u such that (u, s) belongs to a small subset of $\mathbb{Z}_q$ for sure. More precisely, the samples follow an (A, ℓ) distribution where A is of size $q - r + 1$ and $\ell = (s, \cdot)$. Then we apply Theorem 3 and some easy other steps to determine s.

Paper organization: In Section 2, we provide the algorithm for $\mathrm{LFS}(q, n, d)$ and prove Theorem 3. In Section 3 we propose a Fourier sampling based algorithm for $\mathrm{HMS}(q, n, r)$ and prove Theorem 6. Finally, in Section 4 we reduce $\mathrm{HMS}(q, n, r)$ to $\mathrm{LFS}(q, n, q - r + 1)$ and prove Theorem 5. We provide a complete proof for Theorem 3 here, the proofs for the other results are given in the full version [11].

2 An algorithm for LFS

Let p be an (A, ℓ)-distribution on $\mathbb{F}_q^n$, where $|A| = d$. We define α_p as the element $\alpha \in A$ for which $\Pr[\ell(u) = \alpha]$ is maximal (breaking a tie arbitrarily). We start the proof with our main technical Lemma 8 which links p to the space of n-variable polynomials of degree d.

The proof of Lemma 8 requires the following variant of the Schwartz-Zippel lemma [21, 19] (proved in [11]) where the polynomial $g(x)$ is not divisible by a linear function $\ell(x)$ and the samples are drawn from an affine subspace $V_\alpha = \{u \in \mathbb{Z}_q^n : \ell(u) = \alpha\}$ for a fixed $\alpha \in \mathbb{F}_q$.

▶ **Lemma 7.** *Let $g(x_1 \ldots, x_n)$, be a degree d polynomial in $\mathbb{F}_q[x_1, \ldots, x_n]$ that is not divisible by $\ell(x_1, \ldots, x_n) - \alpha$ where $\alpha \in \mathbb{F}_q$ and $\ell(x_1, \ldots, x_n)$ is a nonzero homogeneous linear polynomial. Let $u = (\beta_1, \ldots, \beta_n)$ be sampled uniformly at random from the affine subspace $V_\alpha = \{u \in \mathbb{Z}_q^n : \ell(u) = \alpha\}$, then $\Pr_{u \sim V_\alpha}[g(u) = 0] \leq \frac{d}{q}$.*

▶ **Lemma 8.** *Let $N = \Omega\left(\binom{n+d}{d} d^2 \log \binom{n+d}{d}\right)$ and let $u_1, \ldots, u_N$ be sampled independently from an (A, ℓ)-distribution on $\mathbb{F}_q^n$, where $|A| = d < q$. Then with probability at least $1/2$, every polynomial $g(x_1, \ldots, x_n)$ over $\mathbb{F}_q^n$ of degree at most d, for which $g(u_i) = 0$ for $i = 1, \ldots, N$, is divisible by $\ell(x_1, \ldots, x_n) - \alpha_p$.*

Proof. For $j = 0, \ldots, N$ we set P_j to be the set of polynomials in $\mathbb{F}_q[x_1, \ldots, x_n]$ of degree at most d which take zero value on the first j samples:

$P_j = \{g(x_1, \ldots, x_n) : \deg g \leq d \text{ and } g(u_i) = 0 \text{ for } i = 1, \ldots, j\}$.

In particular, P_0 is the set of all polynomials of degree at most d. We consider P_0 as a vector space of dimension $\binom{n+d}{d}$ over $\mathbb{F}_q$. Since, for $u \in \mathbb{F}_q^n$, the map $g \mapsto g(u)$ is linear on $\mathbb{F}_q[x_1, \ldots, x_n]$, we conclude that $P_0, \ldots, P_N$ is a non-increasing sequence of subspaces of P_0.

Set $\pi = \Pr[\ell(u) = \alpha_p]$, and observe that $\pi \geq \frac{1}{d}$. Let P' be the set of polynomials from P_0 which are divisible by $\ell(x_1, \ldots, x_n) - \alpha_p$. Then an equivalent way to state the lemma is that $P_N \subset P'$, with probability at least $1/2$.

We first claim that, for every $j = 1, \ldots, N$,

$$\Pr[P_j = P_{j-1} | P_{j-1} \not\subseteq P'] \leq 1 - \frac{1}{d(d+1)}. \tag{1}$$

In order to prove this bound, we note that the condition $P_{j-1} \not\subseteq P'$ means that there exists a non zero $g \in P_{j-1} \setminus P'$. Fix such a g. The event $P_j = P_{j-1}$ is equivalent to $f(u_j) = 0$, for all $f \in P_{j-1}$. Therefore

$$\Pr[P_j = P_{j-1} | P_{j-1} \not\subseteq P'] \leq \Pr[\forall f \in P_{j-1},\ f(u_j) = 0] \leq \Pr[g(u_j) = 0].$$

The probability that $g(u_j) = 0$ can be bounded as follows:

$$\begin{aligned}\Pr[g(u_j) = 0] &\leq \Pr[g(u_j) = 0 | \ell(u_j) \neq \alpha_p] \cdot (1 - \pi) + \Pr[g(u_j) = 0 | \ell(u_j) = \alpha_p] \cdot \pi \\ &\leq (1 - \pi) + \pi \frac{d}{q}.\end{aligned}$$

The first inequality follows simply by decomposing the event $g(u_j) = 0$ according to whether $\ell(u_j)$ is different from, or equal to α_p. In the second case, which happens with probability π, Lemma 7 is applicable and it states that $g(u_j) = 0$ with probability at most d/q. This explains the second inequality. Using $\pi \geq 1/d$ and $q \geq d + 1$, a simple calculation gives

$1 - \pi + \pi \frac{d}{q} \leq 1 - \frac{1}{d(d+1)}$,

from which the inequality (1) follows.

Algorithm 1 Algorithm for LFS(q, n, d).

Require: A set $A \subset \mathbb{F}_q$ of cardinality d and a sequence of N elements $u_1, \ldots, u_N$ from $\mathbb{F}_q^n$.

1. Find a nonzero polynomial $g(x_1, \ldots, x_n)$ of degree at most d over $\mathbb{F}_q$, if it exists, such that $g(u_i) = 0$ for $i = 1, \ldots, N$.
2. Compute the linear factors of g.
3. Find a linear factor f of g and a nonzero element $\gamma \in \mathbb{F}_q$, if exist, such that $\gamma(f(u_i) - f(0)) \in A$, for $i = 1, \ldots, N$. Return the linear function $\gamma(f(x_1, \ldots, x_n) - f(0))$.

We can use the conditional probability in (1) to upper bound the probability of the event that $P_{j-1} \not\subseteq P'$ and $P_j = P_{j-1}$ hold simultaneously. But if $P_j = P_{j-1}$ then $P_{j-1} \subseteq P'$ is equivalent to $P_j \subseteq P'$, therefore we can infer, for every $j = 1, \ldots, N$ that $\Pr[P_j \not\subseteq P' \text{ and } P_j = P_{j-1}] \leq 1 - \frac{1}{d(d+1)}$.

Iterating the above argument k-times, we obtain, for every $k \leq N$ and $j \leq N - k + 1$,

$$\Pr[P_{j+k-1} \not\subseteq P' \text{ and } P_{j+k-1} = P_{j-1}] \leq \left(1 - \frac{1}{d(d+1)}\right)^k. \tag{2}$$

Indeed, as before, we can bound the probability on the left hand side by the conditional probability $\Pr[P_{j+k-1} = P_{j-1} | P_{j+k-1} \not\subseteq P']$. Under the condition $P_{j+k-1} \not\subseteq P'$, there exists a non zero $g \in P_{j+k-1} \setminus P'$, and we fix such a g. Then

$$\begin{aligned}\Pr[P_{j+k-1} \not\subseteq P' \text{ and } P_{j+k-1}] &\leq \Pr[g(u_{j+i}) = 0, \text{ for } i = 0, \ldots, k-1] \\ &\leq \prod_{i=0}^{k-1} \Pr[g(u_{j+i}) = 0] \leq (1 - \frac{1}{d(d+1)})^k,\end{aligned}$$

where for the second inequality we used that the samples u_{j+i} are independent.

Taking $k = \Omega(d^2 \log \binom{n+d}{d})$, $N = (\binom{n+d}{d} + 1)k$ and $j = mk + 1$, for $m = 0, 1, \ldots, \binom{n+d}{d}$, in inequality (2), we get $\Pr[P_{(m+1)k} \not\subseteq P' \text{ and } P_{(m+1)k} = P_{mk}] \leq \frac{1}{2}\binom{n+d}{d}^{-1}$.

For the complement of the union of these $\binom{n+d}{d} + 1$ events, we derive then

$$\Pr[\bigcap_{m=0}^{\binom{n+d}{d}} \Big(P_{(m+1)k} \subseteq P' \text{ or } P_{(m+1)k} \subset P_{mk}\Big)] \geq \frac{1}{2}.$$

If $P_{(m+1)k} \subset P_{mk}$ for some m, then $\dim(P_{(m+1)k}) < \dim(P_{mk})$. We can not have simultaneously $\dim(P_{(m+1)k}) < \dim(P_{mk})$, for $m = 0, 1, \ldots, \binom{n+d}{d}$, because otherwise $\dim(P_N)$ would be negative. Therefore, with probability at least $1/2$, $P_{(m+1)k} \subseteq P'$, for some $m \leq \binom{n+d}{d}$, implying $P_N \subseteq P'$. ◀

We now present an algorithm for LFS(q, n, d) and show that it solves the problem efficiently when the input contains a polynomially large number of samples $u_1, \ldots, u_N \in \mathbb{F}_q^n$ from an (A, ℓ)-distribution, with $|A| = d$ constant.

▶ **Theorem 9.** *There is a randomized implementation of* **Algorithm 1** *which runs in time polynomial in* $\log q, \binom{n+d}{d}$ *and* N. *Moreover, when* $u_1, \ldots, u_N$ *are independent samples from an* (A, ℓ)*-distribution on* $\mathbb{F}_q^n$ *where* $|A| = d$ *and* $N = \Omega\left(\binom{n+d}{d} d^2 \log \binom{n+d}{d}\right)$, *then it finds successfully* ℓ *up to a constant factor with probability at least* $1/2$.

Proof. We first describe the randomized implementation with the claimed running time. Throughout the proof by polynomial time we mean time polynomial in $\log q$, $\binom{n+d}{d}$ and N. For Step 1, we consider the $\binom{n+d}{d}$ dimensional vector space of n-variable polynomials over $\mathbb{F}_q$ of degree at most d. The system of requirements $g(u_i) = 0$, for $i = 1, \ldots, N$, is equivalent to a system of N homogeneous linear equations for the $\binom{n+d}{d}$ coefficients of g, where in the ith equation, the coefficients of the variables are the values of the monomials taken at u_i. Therefore a solution, if it exists, can be computed in polynomial time using standard linear algebra.

We use Kaltofen's algorithm [12] (the finite field case is dealt with explicitly in [20]) to find the irreducible factors of g. It is a Las Vegas randomized algorithm, and it runs in polynomial time given the representation of the input polynomial as a list of all coefficients. We can then easily select the linear factors out of the irreducible factors, therefore Step 2 can also be done in polynomial time.

For Step 3, note that g has at most $d \leq n$ linear factors, therefore it is enough to see that each individual factor f can be dealt with in polynomial time. This can be done as follows. If $f(u_i) = f(0)$ for every i, then an appropriate γ can be found if and only if $0 \in A$. Indeed, if $0 \in A$ then any nonzero γ satisfies the condition, while otherwise no satisfying γ exists. Otherwise, pick any i such that $\beta = f(u_i) - f(0) \neq 0$ and try $\gamma = \alpha/\beta$ for every $\alpha \in A$.

We now turn to the proof of correctness of the algorithm when the samples come from an (A, ℓ)-distribution. As $\prod_{\alpha \in A} (\ell(u_i) - \alpha) = 0$, for every i, the algorithm finds a nonzero polynomial g in Step 1. By Lemma 8, with probability at least $1/2$, every polynomial of degree at most d, which is zero on u_i, for $i = 1, \ldots, N$, is divisible by $\ell(x_1, \ldots, x_n) - \alpha_p$. Assume that this is the case. Then, in particular, g has a linear factor $f(x)$ which is a constant multiple of $\ell(x) - \alpha_p$, that is $f(x) = \beta(\ell(x) - \alpha_p)$, for some non zero $\beta \in \mathbb{F}_q$. It is easy to check that for $\gamma = \beta^{-1}$, we have $\gamma(f(x) - f(0)) = \ell(x)$, and therefore $\gamma(f(u_i) - f(0)) \in A$, for $i = 1, \ldots, N$. Thus the algorithm in its last step will find successfully and return a linear function $\ell'(x)$ such that $\ell'(u_i) \in A$, for every i.

To finish the proof, we claim that $\ell'(x)$ is a constant multiple of $\ell(x)$. The polynomial $h(x_1, \ldots, x_n) = \prod_{\alpha \in A}(\ell'(x_1, \ldots, x_n) - \alpha)$ is zero on every u_i and hence, by our assumption, $h(x_1, \ldots, x_n)$ is divisible by $\ell(x_1, \ldots, x_n) - \alpha_p$. Then, as $\mathbb{F}_q[x_1, \ldots, x_n]$ is a unique factorization domain, there exists $\alpha \in A$ such that $\ell'(x_1, \ldots, x_n) - \alpha$ is a scalar multiple of $\ell(x_1, \ldots, x_n) - \alpha_p$, implying the claim. ◀

Theorem 3 is an immediate consequence of this result. For constant d we have the following corollary.

▶ **Corollary 10.** *There is a randomized algorithm that solves* $\mathrm{LFS}(q, n, d)$ *for constant* d *with sample complexity* $\mathrm{poly}(n)$ *and running time* $\mathrm{poly}(n, \log q)$.

We next present our algorithms for $\mathrm{HMS}(q, n, r)$, we first give a basic Fourier sampling based algorithm in section 3 and then an algorithm that reduces $\mathrm{HMS}(q, n, r)$ to $\mathrm{LFS}(q, n, q-r+1)$ in section 4.

3 Fourier sampling algorithm for $\mathrm{HMS}(q, n, r)$

We first describe briefly the standard pre-processing procedure for $\mathrm{HMS}(q, n, r)$. starting with the uniform superposition, append a register consisting of l qubits, initialized to 0 and query the oracle for f_s to obtain,

$$\frac{1}{\sqrt{q^n r}} \sum_{v \in \mathbb{Z}_q^n} \sum_{h \in H} |v\rangle\,|h\rangle \to \frac{1}{\sqrt{q^n r}} \sum_{v \in \mathbb{Z}_q^n} \sum_{h \in H} |v\rangle\,|h\rangle\,|f_s(v, h)\rangle\,.$$

The last l qubits are then measured to obtain the state,

$$\psi_s^w := \frac{1}{\sqrt{r}} \sum_{h \in H} |w + hs\rangle\,|h\rangle\,,$$

where $w \in \mathbb{Z}_q^n$ is uniformly random. This w is the unique element of $\mathbb{Z}_q^n$ such that the measured value for the function f_s equals $f(w)$.

It is standard to then apply the quantum Fourier transform on $\mathbb{Z}_q^n$ to the states ψ_s^w and to measure the first register to obtain tuples (u, ϕ_s^u) where $u \in \mathbb{Z}_q^n$ is uniformly random and $\phi_s^u := \frac{1}{\sqrt{r}} \sum_{h \in H} \omega^{(u,hs)} |h\rangle$. We therefore assume without loss of generality that the quantum input for $\mathrm{HMS}(q, n, r)$ are N samples of the form (u, ϕ_s^u) for uniformly random $u \in \mathbb{Z}_q^n$.

We next give the Fourier sampling based algorithm for $\mathrm{HMS}(q, n, r)$ and prove Theorem 6. The basic idea for the algorithm is to consider the input state $\phi_s^u = \frac{1}{\sqrt{r}} \sum_{h \in H} \omega^{(u,hs)} |h\rangle$ for $\mathrm{HMS}(q, n, r)$, as an approximation to the state $\kappa_s^u := \frac{1}{\sqrt{q}} \sum_{h=0}^{q-1} \omega^{(u,hs)} |h\rangle$. The inner product between the two states is $\phi_s^{u\dagger} \cdot \kappa_s^u = \frac{1}{\sqrt{qr}} \sum_{h \in H} 1 = \sqrt{r/q}$.

The inverse Fourier transform on $\mathbb{Z}_q$, when applied to κ_s^u gives $|(u, s)\rangle$. If we could determine the inner products $|(u, s)\rangle$ for a set of n linearly independent u_i for prime q, then s can be determined by solving a system of linear equations. More generally, in order to make this approach work k should be large enough so that the u_i generate $\mathbb{Z}_q^n$, in this case the secret s can be recovered from the inner products using linear algebra. In fact, the following result from [16] shows that the additive group $\mathbb{Z}_q^n$ is generated by $k = n + O(1)$ random elements of $\mathbb{Z}_q^n$ with constant probability.

▶ **Fact 11.** *[16] Let G be a finite abelian group with a minimal generating set of size r. The expected number of elements chosen independently and uniformly at random from G such that the chosen elements generate G is at most $r + \sigma$ where $\sigma < 2.12$ is an explicit constant.*

The above fact holds for any abelian group, for the special case of $\mathbb{Z}_q^n$ we have $r = n$ and the constant σ can be taken to be 1 [1]. We therefore have that $k = 2n + O(1)$ random elements of $\mathbb{Z}_q^n$ generate the additive group $\mathbb{Z}_q^n$ with constant probability.

If we apply the Fourier transform to each $\phi_s^{u_i}$, with probability $(r/q)^{k/2}$ we obtain the scalar products of s with the members of a generating set for $\mathbb{Z}_q^n$. The answer s may be verified by repeating the experiment for $\mathrm{poly}(n)(q/r)^{k/2}$ trials and finding the most frequently occurring solutions over the different trials.

▶ **Theorem 6.** *There is a quantum algorithm that solves $\mathrm{HMS}(q, n, r)$ with high probability in time $O(\mathrm{poly}(n)(\frac{q}{r})^{n+O(1)})$.*

We next show that the above algorithm runs in time $\mathrm{poly}(n)$ for parameters q, r such that $\frac{r}{q} = 1 - \Omega(\frac{\log n}{n})$. For this choice of parameters, we can bound the factor $(\frac{q}{r})^{n+O(1)}$ in the running time bound above as follows,

$$\left(\frac{r}{q}\right)^{n+O(1)} \geq \left(1 - \frac{c_1 \log n}{n}\right)^{c_2 n + c_3} \geq e^{-c \log n} = n^{-O(1)}$$

where c, c_1, c_2 are suitable constants. We therefore have,

▶ **Corollary 12.** *If $\frac{r}{q} = 1 - \Omega(\frac{\log n}{n})$, then there is a quantum algorithm that solves $\mathrm{HMS}(q, n, r)$ with high probability in time $\mathrm{poly}(n)$.*

4 Reducing $\mathrm{HMS}(q, n, r)$ to $\mathrm{LFS}(q, n, q - r + 1)$

In this section we assume that q is a prime number and work over the field $\mathbb{F}_q$. Recall that the input for $\mathrm{HMS}(q, n, r)$ is a collection of samples of vector-state pairs (u, ϕ_s^u) where u is a uniformly random vector from $\mathbb{F}_q^n$, and $\phi_s^u = \frac{1}{\sqrt{r}} \sum_{h \in H} \omega^{(u,s)h} |h\rangle$. For $t \in \mathbb{F}_q$ define the state $\mu_t := \frac{1}{\sqrt{r}} \sum_{h \in H} \omega^{ht} |h\rangle$, so that $\phi_s^u = \mu_{(u,s)}$.

The approach in Section 3 recovers the inner product (u_i, s) for $O(n)$ random vectors u_i and then uses Gaussian elimination to determine s with high probability. However, the μ_t's are only nearly orthogonal to each other, so the measurement in Section 3 may fail to recover the correct value of (u, s) with probability too large for our purposes.

A particularly interesting case is when $q = \text{poly}(n)$ and $c = q - r$ is a constant for which we provide a polynomial time algorithm in Corollary 20. In this case, the error probability for the measurement in Section 3 is $1 - r/q = c/q$, that is there are a constant expected number of errors for every q samples. If $q = O(n^\alpha)$ for $\alpha < 1$ then there are $O(n^{1-\alpha})$ errors in expectation for every n samples. There are no known polynomial time algorithms for recovering the secret $s \in \mathbb{Z}_q^n$ from a system of n linear equations where an $O(n^{1-\alpha})$ fraction of the equations are incorrect for a constant α.

Instead, we reduce $\text{HMS}(q, n, r)$ to $\text{LFS}(q, n, d)$ with $d = (q-r)+1$, $A = \{r-1, \ldots, q-1\}$ and the linear function $\ell(\cdot)$ given by $\ell(x) = (s, x)$. We then use the Algorithm 1 to recover a scalar multiple of $s_0 = \lambda s$. Further, we show that the scalar λ can be recovered efficiently.

The reduction performs a quantum measurement on ϕ_s^u to determine if (u, s) belongs to $A = \{r-1, \ldots, q-1\}$. We discard the u's which do not belong to A, and also some of the u's such that $(u, s) \in A$ to obtain samples from an (A, ℓ) distribution. We next provide a sketch of the reduction from $\text{HMS}(q, n, r)$ to $\text{LFS}(q, n, d)$, the reduction is analyzed over the next few subsections and a more precise statement is given in Proposition 17.

Let V be the hyperplane spanned by $\mu_0, \ldots, \mu_{r-2}$. Let (u, ϕ_s^u) be a pair from the input samples. We perform the measurement on ϕ_s^u according to the decomposition of $\mathbb{C}^r = V \oplus V^\perp$, and retain u if and only if the result of the measurement is 'in $V^\perp$. Otherwise we discard u. An efficient implementation of the measurement in $(V, V^\perp)$ is given later in this section.

Observe that measuring a state μ_j 'in $V^\perp$ is only possible if $\mu_j \notin V$, in particular $j \notin \{0, \ldots, r-2\}$. Thus if we measure $\phi_s(u)$ 'in $V^\perp$ we can be sure that (s, u) is in $A = \{r-1, \ldots, q-1\}$. We only keep u from a sample pair (u, τ) if this measurement, applied to the state τ, results 'in $V^\perp$'. The u's that are retained are samples from an (A, ℓ) distribution over $\mathbb{F}_q^n$.

We bound the probability of retaining a sample pair (u, ϕ_s^u) for this procedure. We bound the success probability for the special case when $(s, u) = r - 1$. As u is uniformly random over $\mathbb{F}_q^n$ the value of (s, u) is uniformly distributed over $\mathbb{Z}_q$, this bound therefore suffices for our purposes.

In the standard basis $|h\rangle$ $(h \in H)$, the vector μ_t has entry $\frac{1}{\sqrt{r}}\omega^{ht}$ in the h-th position. Let $A \in \mathbb{C}^{r \times r}$ be the matrix with rows from the collection $\{\mu_t : 0 \le t \le r-1\}$, that is

$$A = \frac{1}{\sqrt{r}} \begin{pmatrix} 1 & \omega^{h_1} & \cdots & \omega^{h_1(r-1)} \\ 1 & \omega^{h_2} & \cdots & \omega^{h_2(r-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{h_r} & \cdots & \omega^{h_r(r-1)} \end{pmatrix}, \tag{3}$$

where $h_1, \ldots, h_r$ are the elements of H, say, in increasing order. The matrix A is $\frac{1}{\sqrt{r}}$ times a Vandermonde matrix and as such, it is well known that it has determinant $r^{-r/2} \prod_{j<i\le r}(\omega^{h_i} - \omega^{h_j})$. In particular, the states $\mu_0, \ldots, \mu_{r-1}$ are linearly independent. With a more careful analysis, we show in Lemma 13 below that $\det A$ is sufficiently far from zero.

▶ **Lemma 13.** *Let $c = q - r$ and A be the matrix in (3) then $|\det A^* A| = \Omega(q^{-c^2}\left(\frac{q}{r}\right)^r)$.*

Using the above lemma, we bound the probability of retaining u if $(u, s) = r - 1$. It might be possible to prove similar bounds for other values of $(s, u) \in A$, however the bound for the particular value $r - 1$ suffices for our purpose.

▶ **Lemma 14.** *The $(V, V^\perp)$-measurement applied to a state of the form μ_t, returns "in V" with probability 1 if $t \in \{0, \ldots, r-2\}$, while for $t \in \{r-1, \ldots, q-1\}$, the probability that "in $V^\perp$" is returned depends only on t and is $\Omega\left(q^{-c^2}\left(\frac{q}{r}\right)^r\right)$ for $t = r-1$.*

We describe next the implementation of the $(V, V^\perp)$ measurement that acts on $O(\log q)$ qubits. Using universality constructions and the Solovay-Kitaev theorem, it is well known that an arbitrary unitary operator can be approximated using an exponential number of elementary gates in the number of qubits.

▶ **Fact 15.** *[14] An arbitrary unitary operation U on t qubits can be simulated to error ϵ using $O(t^2 4^t \log^c(t^2 4^t/\epsilon))$ elementary gates.*

The ability to implement an arbitrary unitary operation on $\log q$ qubits implies the ability to perform the measurement $(W, W^\perp)$ for an arbitrary subspace $W \subset \mathbb{C}^q$.

Denote the quantum state corresponding to unit vector $w \in \mathbb{C}^q$ as $|w\rangle := \sum_{i=1}^q w_i |i\rangle$. Let k be the dimension of W and let $w_1, w_2, \cdots, w_k$ be an orthonormal basis for W. Let U_W be a unitary operation that maps the standard basis vectors $|i\rangle \to |w_i\rangle$. Then the measurement in $(W, W^\perp)$ on state $|\phi\rangle$ can be implemented by first computing $U_W^{-1}|\phi\rangle$ and then measuring in the standard basis. The state $|\phi\rangle$ belongs to W if and only if the result of measurement in the standard basis belongs to the set $\{1, 2, \cdots, k\}$. As the $(V, V^\perp)$ measurement is on $\log q$ qubits, by Fact 15 we have,

▶ **Claim 16.** *The measurement $(V, V^\perp)$ can be implemented to precision $1/q^{O(1)}$ in time $\widetilde{O}(q^2)$.*

The implementation of the $(V, V^\perp)$ measurement above shows that the sampling procedure can be performed efficiently. The procedure yields a sample from an (A, ℓ) distribution with $|A| = (q-r)+1$ when the measurement outcome is $V^\perp$. By Lemma 14 the outcome $V^\perp$ occurs with probability $\Omega\left(q^{-c^2}\left(\frac{q}{r}\right)^r\right)$ if $(u, s) = r-1$. As u is uniformly random on $\mathbb{F}_q^n$ at least a $\Omega\left(q^{-c^2-1}\left(\frac{q}{r}\right)^r\right)$ fraction of the samples are retained. We therefore have the following proposition,

▶ **Proposition 17.** *There is a quantum procedure that that runs in time $\widetilde{O}(q^2)$, and given a pair (u, ϕ_s^u) where $u \in \mathbb{Z}_q^n$ is uniformly random and $\phi_s^u = \mu_{(u,s)}$, with probability at least $O\left(q^{-c^2-1}\left(\frac{q}{r}\right)^r\right)$ returns a sample from a (A, ℓ) distribution with $|A| = c+1$ and $\ell(x) = (s, x)$.*

In order to solve $\mathrm{HMS}(q, n, r)$ given a scalar multiple $s_0 = \lambda s$ found using Theorem 9, we need to find the scalar λ. We show that using $O(q)$ further input pairs we can find the value of λ using a simple trial and error procedure given by the following lemma.

▶ **Lemma 18.** *Given $t \in \mathbb{F}_q$ and a state $\tau \in \mathbb{C}^r$, there is a quantum procedure that returns YES with probability 1 if $\tau = \mu_t$, while if $\tau = \mu_{t'}$ for some $t' \in \mathbb{F}_q \setminus \{t\}$, it returns YES with probability at most $p := \begin{cases} 1/4 \text{ if } q < 3r/2 \\ 1 - O(1/q) \text{ otherwise.} \end{cases}$*

Combining the results proved in this section with Algorithm 1, we next obtain a quantum algorithm for $\mathrm{HMS}(q, n, r)$ for the case of prime q.

Theorem 9 shows that given $N = \Omega(\binom{n+d}{d} d^2 \log(\binom{n+d}{d}))$ samples from an (A, ℓ) distribution, a scalar multiple of the function ℓ can be found with constant probability. Proposition 17 above shows that the expected time to obtain N samples from the (A, ℓ) distribution is

$\widetilde{O}(Nq^{c^2+3})$ where we used that each measurment requires time $\widetilde{O}(q^2)$ and ignored the factor $(r/q)^r < 1$. The number of samples and the time required for determining the scalar λ in Lemma18 are negligible compared to these quantities. We therefore have the following theorem,

▶ **Theorem 19.** *Let q be a prime and let $c = q - r$. Then there is a quantum algorithm which solves* $\mathrm{HMS}(q, n, r)$ *with sample complexity $\widetilde{O}(n^{c+1}q^{c^2+1})$ and in time $\widetilde{O}(n^{c+1}q^{c^2+3})$.*

The algorithm runs in time polynomial in $n, \log q$ for the case when $q = \mathrm{poly}(n)$. We therefore we have the following corollary,

▶ **Corollary 20.** *Let $q = \mathrm{poly}(n)$ be a prime number and $c = q - r$ be a constant, then there is an efficient quantum algorithm for* $\mathrm{HMS}(q, n, r)$.

References

1. Vincenzo Acciaro. The probability of generating some common families of finite groups. *Utilitas Math.*, 49:243–254, 1996.
2. Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In *Proceedings of the 38th International Colloquium on Automata, Languages and Programming ICALP, Zurich, Switzerland*, pages 403–415. Springer, 2011.
3. Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003.
4. Andrew M. Childs, Robin Kothari, Maris Ozols, and Martin Roetteler. Easy and hard functions for the boolean hidden shift problem. In Simone Severini and Fernando G. S. L. Brandão, editors, *8th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2013, May 21-23, 2013, Guelph, Canada*, volume 22 of *LIPIcs*, pages 50–79. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013. doi:10.4230/LIPIcs.TQC.2013.50.
5. Andrew M. Childs and Wim van Dam. Quantum algorithm for a generalized hidden shift problem. In Nikhil Bansal, Kirk Pruhs, and Clifford Stein, editors, *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 1225–1232. SIAM, 2007. URL: http://dl.acm.org/citation.cfm?id=1283383.1283515.
6. Thomas Decker, Gábor Ivanyos, Raghav Kulkarni, Youming Qiao, and Miklos Santha. An efficient quantum algorithm for finding hidden parabolic subgroups in the general linear group. In *International Symposium on Mathematical Foundations of Computer Science*, pages 226–238. Springer, 2014.
7. Mark Ettinger and Peter Høyer. On quantum algorithms for noncommutative hidden subgroups. *Advances in Applied Mathematics*, 25:239–251, 2000. arXiv:arXiv:quant-ph/9807029.
8. Katalin Friedl, Gábor Ivanyos, Frédéric Magniez, Miklos Santha, and Pranab Sen. Hidden translation and translating coset in quantum computing. *SIAM Journal on Computing*, 43(1):1–24, 2014. preliminary version in STOC 2003.
9. Dmitry Gavinsky, Martin Roetteler, and Jérémie Roland. Quantum algorithm for the boolean hidden shift problem. In Bin Fu and Ding-Zhu Du, editors, *Computing and Combinatorics - 17th Annual International Conference, COCOON 2011, Dallas, TX, USA, August 14-16, 2011. Proceedings*, volume 6842 of *Lecture Notes in Computer Science*, pages 158–167. Springer, 2011. doi:10.1007/978-3-642-22685-4_14.
10. Gábor Ivanyos. On solving systems of random linear disequations. *Quantum Information & Computation*, 8(6):579–594, 2008. URL: http://www.rintonpress.com/xxqic8/qic-8-67/0579-0594.pdf.

11 Gábor Ivanyos, Anupam Prakash, and Miklos Santha. On learning linear functions from subset and its applications in quantum computing. *arXiv*, 2018. arXiv:1710.02581.

12 Erich Kaltofen. Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorization. *SIAM J. Comput.*, 14(2):469–489, 1985. doi:10.1137/0214035.

13 Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing*, 35(1):170–188, 2005. arXiv:quant-ph/0302112.

14 Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

15 Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 333–342. ACM, 2009.

16 Carl Pomerance. The expected number of random elements to generate a finite abelian group. *Periodica Mathematica Hungarica*, 43(1):191–198, Aug 2002. doi:10.1023/A:1015250102792.

17 Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

18 Martin Roetteler. Quantum algorithms for abelian difference sets and applications to dihedral hidden subgroups. In Anne Broadbent, editor, *11th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC, September 27-29, 2016, Berlin, Germany*, volume 61 of *LIPIcs*, pages 8:1–8:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.TQC.2016.8.

19 Jacob T Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM (JACM)*, 27(4):701–717, 1980.

20 Joachim von zur Gathen and Erich Kaltofen. Polynomial-time factorization of multivariate polynomials over finite fields. In *International Colloquium on Automata, Languages, and Programming*, pages 250–263. Springer, 1983.

21 Richard Zippel. Probabilistic algorithms for sparse polynomials. *Symbolic and algebraic computation*, pages 216–226, 1979.